

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practice in the
Company

2019

Daniel Soukenka

Zadání bakalářské práce

Student:

Daniel Soukenka

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: ISSA CZECH s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

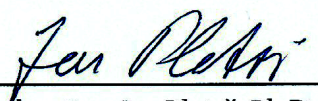
Vedoucí bakalářské práce: **Ing. Radim Bača, Ph.D.**

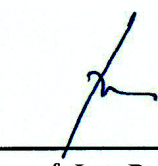
Konzultant bakalářské práce: Ing. Michal Kolesár, Ph.D.

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019




doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry


prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 26. dubna 2019

.....
Soukenka

Souhlasím se zveřejněním této bakalářské/diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských/magisterských programech VŠB-TU Ostrava.

V Ostravě 24. dubna 2019



ISSA

ISSA CZECH s.r.o.
Hrušovská 3203/13a
702 00 Ostrava
DIČ: CZ25381920
www.issa.cz



V první řadě bych chtěl velmi poděkovat společnosti ISSA CZECH s.r.o. za příležitost absolvování odborné praxe a získání četných zkušeností a mému konzultantovi bakalářské práce Ing. Michalu Kolesárovi, Ph.D., který mi byl vždy jako první nápomocen při řešení vyvstávších problémů. Dále bych rád poděkoval Darkovi Gomolovi za spolupráci a rady ohledně práce ve firemním frameworku. Děkuji vedoucímu bakalářské práce Ing. Radimu Bačovi Ph.D. za doporučení týkající se zpracování této práce a Lukáši Buroňovi, který mi poskytl rady v mých prvních dnech ve firmě.

Abstrakt

Tato bakalářská práce popisuje mé působení během absolvování odborné praxe ve firmě ISSA CZECH s.r.o. Podrobněji se v ní zabývám úkoly, které jsem obdržel a jejich řešeními. Těmi byly tvorba webových stránek a vytvoření firemního intranetu. Představuji mnou využitě technologie a věnuji se i obecně celému průběhu mé práce včetně hodnocení konceptu odborných praxí.

Klíčová slova

ISSA CZECH s.r.o., individuální odborná praxe, bakalářská práce, CMS, webové stránky, intranet, PHP framework, webová aplikace, informační systém

Abstract

This bachelor's thesis describes my work during the individual professional practice in the company ISSA CZECH s.r.o. I deal in more detail with the tasks I have received and their solutions. These were the creation of web pages and the creation of a company's intranet. I present the technologies used by me and I devote myself to the whole process of my work, including the evaluation of the concept of professional practice.

Key Words

ISSA CZECH s.r.o., individual professional practice, bachelor's thesis, CMS, web pages, intranet, PHP framework, web application, information system

Obsah

| | |
|--|-----------|
| Seznam použitých symbolů a zkratk | 9 |
| Seznam ilustrací | 10 |
| Seznam výpisů zdrojového kódu | 11 |
| 1 Úvod | 12 |
| 2 Popis firmy, jejího zaměření a pracovního zařazení studenta | 14 |
| 2.1 Popis firmy | 14 |
| 2.2 Popis pracovního zařazení | 14 |
| 3 Seznam a popis zadaných úkolů v průběhu odborné praxe | 15 |
| 3.1 Webové stránky rukov.cz | 15 |
| 3.1.1 Předmluva | 15 |
| 3.1.2 Specifikace zadání | 15 |
| 3.2 Firemní intranet | 16 |
| 3.2.1 Předmluva | 16 |
| 3.2.2 Důvody vzniku projektu | 16 |
| 3.2.3 Specifikace zadání | 17 |
| 4 Řešení zadaných úkolů | 19 |
| 4.1 Webové stránky rukov.cz | 19 |
| 4.1.1 Použité technologie | 19 |
| 4.1.2 Řešení | 22 |
| 4.2 Firemní intranet | 23 |
| 4.2.1 Použité technologie | 23 |
| 4.2.2 Řešení | 24 |
| 5 Znalosti či dovednosti získané v průběhu studia a uplatněné v průběhu odborné praxe | 29 |
| 6 Znalosti či dovednosti scházející v průběhu odborné praxe | 30 |
| 7 Závěr | 31 |
| 7.1 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení | 31 |
| Literatura | 32 |

Seznam použitých symbolů a zkratek

AJAX - Asynchronous JavaScript and XML
API - Application Programming Interface
ASP.NET - Active Server Pages
CRUD - Create, Read, Update, Delete
CMS - Content Management System
CSS - Cascading Style Sheets
CVS - Concurrent Version System
DOM - Document Object Model
HTML - Hypertext Markup Language
HTTP - Hypertext Transfer Protocol
IIS - Internet Information Services
JS - JavaScript
JSON - JavaScript Object Notation
LAMP - Linux, Apache, MySQL, PHP
PDF - Portable Document Format
PHP - PHP: Hypertext Preprocessor
SEO - Search Engine Optimization
SGML - Standard Generalized Markup Language
SQL - Structured Query Language
WAMP - Windows, Apache, MySQL, PHP
XHTML - Extensible Hypertext Markup Language

Seznam ilustrací

| | | |
|---|---|----|
| 1 | Ukázka webu rukov.cz - návrh katalogu produktů | 16 |
| 2 | Ukázka webu rukov.cz - řešení katalogu produktů | 23 |
| 3 | Ukázka z intranetu - filtr iniciály jména | 25 |

Seznam výpisů zdrojového kódu

Výpis 1: Procedura pro vygenerování docházky zaměstnance v daném roce a měsíci . 27

1 Úvod

Tato bakalářská práce podává ucelenou závěrečnou zprávu o mé odborné praxi, kterou jsem vykonával během 3. ročníku bakalářského studia na Fakultě elektrotechniky a informatiky Vysoké školy báňské. Tuto praxi jsem absolvoval ve firmě ISSA CZECH s.r.o. působící na trhu informačních technologií v ostravském regionu. Referuji zde o jejím průběhu - o zadaných úkolech, jejich řešeních, výsledcích a přínosech. Dále zmiňuji mnou nabyté znalosti, dovednosti a zkušenosti.

Pro vykonání bakalářské práce alternativní formou absolvování odborné praxe ve firmě (naproti teoretické práce) jsem se rozhodl z několika důvodů. Tím hlavním a nejdůležitějším byla možnost získat praktické zkušenosti a vůbec představu o tom, jak vývoj informačních systému reálně probíhá v komerční oblasti. Během studia ani svého osobního života jsem totiž dosud neměl příležitost vytvořit si v tomto dostatečný obraz a přehled. Při teoretických či praktických, ale přesto akademických, hodinách různých předmětů jsem si vždy kladl otázky, jak dané znalosti využiji při řešení nějakého reálného projektu. A jak vůbec takový reálný projekt vypadá? Jak vypadá jeho zadání? Jak postupovat a jak postupovat správně při jeho řešení? Odpovědi na tyto otázky mi škola ani nemohla poskytnout, jelikož jediný nezkreslený způsob, jak je získat, je přímo se na takovém projektu podílet.

Dalším aspektem důležitým při rozhodování byla příležitost zařadit se do pracovního procesu způsobem, který nepředpokládal dokonalou znalost všech nástrojů či postupů vývoje. Místo toho poskytuje možnost využít potenciálu a vůle studenta pro začlenění se do chodu organizace, která k danému studentovi přistupuje tolerantně a nápomocně. Tento způsob hodnotím kladně a jedná se o něco, co jsem sám vyhledával. Dalším důvodem pak byla samotná možnost pro zvýšení všeho označovaného v ekonomii jako lidský kapitál. Tedy souboru znalostí, schopností a dovedností jedince.

Konkrétní společnost pro svou praxi jsem vybíral dle několika hledisek, z nichž nejdůležitější bylo její zaměření. Chtěl jsem se programátorsky podílet na vývoji, co největší části nějaké aplikace nebo programu a získat tak v tomto ucelenou představu. Kromě lokality, jsem se dále rozhodoval podle komunikace a celkového vystupování firmy. Také jsem spíše upřednostňoval práci v malé firmě se zdravou, klidnou a pozitivní atmosférou a prostředím.

V úvodní části práce se věnuji popisu firmy, kde jsem získal příležitost praxi vykonat, a mému pracovnímu zařazení v ní. Dále popisuji úkoly, na nichž jsem se pracoval. Prvně čtenáři poskytnu vstupní pohled do problému zadaného úkolu a poté uvádím a vysvětluji mé řešení vzhledem k dostupným a použitým technologiím. Přitom vybírám méně obvyklé či složitější části řešení, které popisuji podrobněji. Prezentuji zde teoretické a praktické znalosti a dovednosti, které jsem získal během studia a při řešení uplatnil, ale také ty, které mi scházely a nabyly jsem jich až v průběhu absolvování. Nakonec bych chtěl celou praxi zhodnotit a sdělit svou úvahu nad konceptem této formy bakalářské práce.

Během praxe jsem pracoval na dvou projektech. Ten první se týkal vytvoření webových stránek pro zákazníka, jímž byla společnost RUKOV Moravia s.r.o. zabývající se kovovýrobou a kovoobráběním. U tohoto úkolu jsem měl možnost získat velké množství poznatků o práci

na projektech a také o mnoha technologiích využívaných při vývoji moderních webových stránek. S většinou těchto technologií jsem se dříve nikdy nesetkal. Jednalo se například o webový framework Bootstrap nebo o firemní CMS software označovaný jako CMS Designer. Tyto technologie blíže popisuji dále v práci.

Mým druhým úkolem byl vývoj intranetu. Zde se jednalo o interní úkol, který mi tak vyšel vstříc s mými časovými možnostmi vzhledem ke studijním povinnostem. Při řešení jsem musel prokázat, jak si dokážu poradit se zadáním většího a komplexního systému na všech jeho vrstvách. Pracoval jsem také na více stupních vývojového procesu software. Při řešení jsem nezačínal, jak se říká, od píky, ale oporou mi byl firemní framework sloužící jako podpora pro rychlejší a pohodlnější vývoj aplikací.

2 Popis firmy, jejího zaměření a pracovního zařazení studenta

2.1 Popis firmy

Jak jsem se zmínil již v úvodní kapitole této práce, má odborná praxe se odehrávala ve firmě ISSA CZECH s.r.o. působící v Ostravě. V té nalezneme okolo 20 zaměstnanců. Díky tomu se zde v průběhu času vyvinulo prostředí s pro firmu charakteristickou atmosférou projevující se v osobnějších vztazích, větší sounáležitostí a důvěrou.

Společnost byla založena roku 1997, kdy nastal prudký rozvoj informačních technologií, zvláště v oblasti Internetu. Název ISSA je zkratkou z "Internet Software and Security Advice", který tímto vypovídá o hlavním poli působnosti společnosti.

Poskytovaná řešení ve sféře webu se týkají webdesignu, webových stránek, e-shopů, SEO nebo internetového marketingu. Vytváří také informační systémy a mobilní aplikace. Webové stránky a informační systémy jsou provozovány ve firemním datovém centru. Dalšími nabízenými službami jsou poradenství a školení v oblasti informačních a komunikačních technologií, správa internetových domén, provoz emailů nebo vzdálená správa počítačů, webů a sítí.[1][2]

2.2 Popis pracovního zařazení

Má pracovní pozice ve společnosti by se dala nejlépe vystihnout jako programátor či vývojář webových stránek a webových aplikací. Jednalo se tedy o aplikace využívající webovou službu Internetu.

V průběhu mé praxe se složitost zadávaných úkolů postupně zvyšovala. Takto jsem se mohl dobře seznámit s prostředím, frameworky, technologiemi a pracovním postupem. Nikterak jsem však neměl pocit, že by se jednalo o úkoly podružné, naopak vycházely z aktuálních potřeb firmy a na obdobných pracovali i mí kolegové.

Na řešení jsem pracoval samostatně a chybějící znalosti jsem doplňoval samostudiem. Po celou dobu jsem se mohl také obracet se svými dotazy na kolegy, kteří mi byli ochotni pomoci. Činil jsem tak především, když jsem měl otázky týkající se firemního frameworku. Rovněž jsem se mohl radit a doplňovat jednotlivá zadání se svým konzultantem ve firmě.

Práce na druhém projektu (firemního intranetu) vyžadovala také jistou součinnost vyplývající z toho, že na řešení úkolu jsme se podíleli společně s dalším kolegou. Často jsme potřebovali na sebe navazovat, takže jsem musel také dodržovat jistá pravidla a zvyklosti týkající se hlavně zápisu zdrojového kódu, se kterými jsem se průběžně seznamoval.

3 Seznam a popis zadanych úkolů v průběhu odborné praxe

Na tomto místě bych chtěl představit úkoly, na kterých jsem po dobu praxe pracoval, a uvést jejich zadání.

3.1 Webové stránky rukov.cz

3.1.1 Předmluva

To první, s čím jsem se na své praxi setkal byl firemní CMS software. Ten mi představil jeden z kolegů. Poté jsem měl možnost se s tímto nástrojem nejprve seznámit, sám si jej vyzkoušet a následně jsem dostal zadání, pro které jsem jej hned využil. Blíže jej popisuji v kategorii použitých technologií pro řešení úkolu.

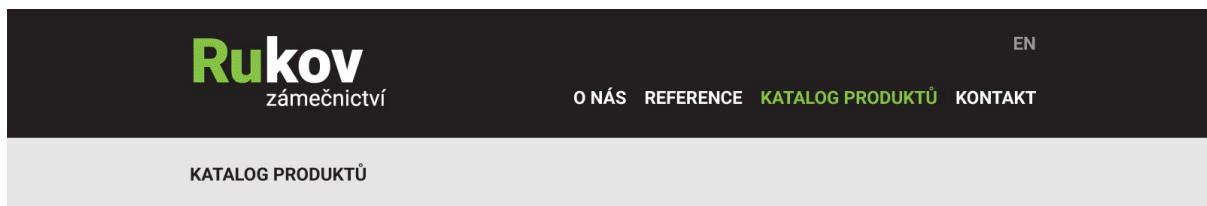
3.1.2 Specifikace zadání

Zadání jsem obdržel v grafické podobě formou několika statických pohledů na výsledné stránky. Cílem bylo vytvořit stejný design a strukturu stránek, jejichž obsah je pak pod správou zákazníka.

Mezi další podmínky patřilo, aby daný web byl plně responzivní, tedy aby se automaticky přizpůsoboval displeji uživatele. Tato podmínka, která se v dnešní době stává standardem při vývoji webových stránek, zajišťuje podporu pro klientelu využívající k prohlížení webu tabletu či mobilních a jiných zařízení. Prohlížení se stává i na malém displeji přívětivé a zásadně ovlivňuje UX (user experience), jinak řečeno uživatelský prožitek.

Struktura webu měla být dle zadání tvořena hlavní stránkou s menu, přes které se zákazník dostane na čtyři další podstránky. Tři z těchto podstránek měly být pouze statické s obsahem, který si již doplní zákazník. Na čtvrté stránce ale požadoval vygenerovaný seznam produktů. Obdobný seznam s typy produktů se měl nacházet také na hlavní stránce. Kromě něj se zde měl vyskytovat velký billboard s vlastním textem. Celý web by měl být dostupný jak v češtině, tak v anglické mutaci.

Design webu byl navržen tematicky k zaměření klienta na odvětví kovovýroby. Mým úkolem bylo tento návrh naimplementovat. Rozložení stránek se skládalo ze záhlaví, ve kterém bylo umístěno logo, menu a přepínač do anglické verze webu. Po samotném obsahu se mělo nacházet zápatí s kontaktními údaji firmy a dalšími informacemi.



Obrázek 1: Ukázka webu rukov.cz - návrh katalogu produktů

3.2 Firemní intranet

3.2.1 Předmluva

Než jsem dostal zadán svůj další úkol, seznámil jsem se s firemním PHP frameworkem pro snadnější vývoj webových aplikací. Ten je vhodně uzpůsoben pro interní systémy firem a dalších zákazníků. Jeden z kolegů mě zaškolil v základních principech tohoto frameworku. Pro jeho komplexnost jsem se však se spousty jeho vlastnostmi obeznamoval i následně v průběhu práce s ním. Dále framework popisuji v části o použitých technologiích při řešení úkolu. Po seznámení jsem obdržel první část zadání intranetu společnosti, ve které jsem odbornou praxi vykonával.

3.2.2 Důvody vzniku projektu

Vzhledem k tomu, že se jednalo o náhradu stávajícího systému, chtěl bych také uvést důvody, které vedly k jeho obměně a vzniku potřeby nového intranetu.

Klíčovým aspektem bylo jeho stáří. Původní verze byla stará přibližně 15 let a za tu šel vývoj webových technologií znatelně dopředu. Jedním z požadavků bylo, aby byl nový intranet responzivní. Kvůli těmto zastaralým systémům pak musí firma obhospodařovat virtuální servery se zastaralou konfigurací operačního systému.

Vývojem prošla ale také samotná firma a některé funkcionality systému přestaly splňovat její potřeby. Mezitím byly samotnou firmou vydány různé standardy a strategie pro vývoj systémů, aby se zabezpečila jejich životnost. To znamená, aby systém zůstal nadále podporovatelný a udržitelný a aby tuto podporu mohl poskytnout i jiný zaměstnanec, který se na vývoji nepodílel. Měl by být také schopný systém dále rozvíjet s ohledem na nové potřeby,

kteře mohou vyplývat např. z legislativního prostředí. Z ekonomického hlediska může nový systém přinést zkvalitnění a urychlení vnitřních procesů, což může vést k úspoře.

3.2.3 Specifikace zadání

Se zadáním intranetu jsem se seznamoval postupně. Mým prvním úkolem bylo vytvořit pomocí frameworku dva číselníky. Jednalo se o evidenci bank a navazující seznam států. Musel jsem zde řešit jak perzistentní uložení dat, tak následnou reprezentaci záznamů v paměti a také jejich zobrazení na prezentační vrstvě webové aplikace. Kromě implementace základních CRUD operací nad číselníky bylo mým úkolem zaměřit se také na filtraci záznamů. Jistou podporu v tomto mi samozřejmě nabídl samotný framework, nicméně se zde vyskytoval i požadavek na poněkud speciální typ filtru, který jsem musel do systému naimplementovat.

Tento filtr umožňuje vylistování bank dle iniciály jejich názvu. Uživatel se následně zobrazuje jako množina abecedně seřazených znaků, přičemž jsou uvedeny jen ty znaky, pod kterými se v databázi skutečně nějaké záznamy nacházejí. Výběr filtrované iniciály uživatel provede kliknutím na znak.

Po dokončení číselníků jsem se již seznámil se zadáním celého intranetu, které bylo opatřeno pohledy na starou verzi a doplněno poznámkami. Na projektu jsem pracoval s dalším kolegou, který již měl hotový modul pro evidenci pracovníků. Pro synchronizaci kódů jsme při vývoji používali systém CVS, který je určen pro správu verzí. Mě dále čekala práce na složitějších modulech s komplexnější logikou, které tvoří podstatu intranetu. Těmi jsou modul stravenky a modul docházky. Ty navázaly na právě dodělanou evidenci zaměstnanců.

Jako první jsem obdržel zadání pro vytvoření části týkající se evidence stravenek. Zde bylo třeba rozlišovat mezi přijatými a vydanými stravenkami, což bylo elementárním kritériem pro většinu následující logiky. Nad rámec vytvoření modulu se základními CRUD operacemi a filtry byly v zadání požadavky na sumaci či hromadné vygenerování a vložení dat. Sumace měla být prováděna na attributech počtu kusů a celkové hodnoty záznamu, která se vypočítala právě dle počtu kusů a hodnoty stravenky. Funkcionalita hromadného vkládání měla za cíl nejprve dle zadaných parametrů (rok a měsíc) vybrat z evidence zaměstnanců vhodné záznamy. Pro tyto pracovníky se vygenerovaly záznamy pro vložení do modulu stravenky a v seznamu se zobrazily uživateli. Ten si mohl ze seznamu vybrat pouze určité záznamy a ty doplnil o požadované hodnoty, které ve vygenerovaných údajích chyběly. Nakonec došlo ke vložení a zaznamenání na perzistentní úložiště.

Dále se v zadání vyskytoval požadavek na emailing. Při vložení nebo úpravě výdajového záznamu došlo k zaslání emailu na adresu daného příjemce. Tímto byl informován o tom, že si stravenky může převzít a poté bylo jeho úkolem převzetí potvrdit v systému. Na tomto místě bylo zapotřebí abych se seznámil s další částí frameworku, totiž se sekcí pro nastavení práv uživatelů. Kromě vytvoření práv na moduly, role zaměstnance a vazeb mezi těmito právy a rolí bylo důležité také vytvořit typ uživatele, který narozdíl od rolí, mohl být využit například pro žádané omezení přístupu zaměstnance pouze ke svým datům a ostatním restrikcím.

Po dokončení této části systému jsem obdržel zadání pro jeho další součást. Kolega měl již hotové číselníky, které byly nutné pro další vývoj, a na které jsem měl navázat. Samotný modul

se skládá z agendy pro evidenci absencí a dalšími dvěma agendami pro správu a přehledné zobrazení docházek pracovníků. Evidence absencí slouží zaměstnancům k ohlášení a zapsání absence, kterou musí následně schválit osoba s odpovídajícími právy. Tato část byla také jako evidence stravenek rozšířena o některé nestandardní vlastnosti. Těmi byl například emailing, kdy po vytvoření nebo úpravě záznamu absence se automaticky zašle zpráva všem uživatelům se speciálním právem umožňujícím schválení absence.

Mým dalším úkolem byl systém pro zaznamenávání docházek pracovníků ve firmě. Zadání bylo neobvyklé zvláště v tom, že se dle něj měly v systému vyskytovat dva pohledy na jednu a tutéž problematiku. Lišily se však ve způsobu zpracování dat a jejich množství. Pod záložkou nazvanou *Docházky jednotlivci* se měly nacházet detailnější údaje o docházce konkrétní osoby pro daný rok a měsíc. Dle těchto parametrů tak bylo potřeba rozlišovat jednotlivé záznamy domény. Naproti tomu v záložce *Docházky souhrn* už rozlišení probíhá pouze dle roku a měsíce, což poskytuje ucelený pohled na denní docházku v měsíci všech zaměstnanců.

Kromě základních operací se v zadání vyskytoval požadavek na hromadné vygenerování a vložení záznamů, které bylo, jako proces, naprosto stejné jako u modulu stravenek, proto bylo namísto se zde zamyslet nad znovupoužitelností kódu, která by zabránila jeho redundancím. Tato funkce se měla vyskytovat v obou záložkách docházek se stejnou logikou. Další požadavek se týkal samotného zobrazení v detailu docházky, neboť ten vyžadoval také promítnutí schválených absencí, aby došlo k zobrazení skutečného a pravdivého přehledu. To ale nebylo vše, zapotřebí bylo ještě vytvořit statistiku. Ta eviduje jak počty dní pro všechny typy absencí, tak také plánované a odpracované hodiny a dny. Kromě toho, že se obě agendy částečně prolínají, mají také svá specifika. U detailu záznamu v souhrnných docházkách to bylo například vyexportování do PDF souboru.

Po dokončení mé práce došlo k vystavení systému na serveru a k jeho testování, kdy mým úkolem zůstalo opravit případné nedostatky, provést drobné úpravy, například grafického charakteru, případně řešit nové požadavky pro další funkcionality. Jedním z takovým požadavků se ukázala být potřeba katalogu svátků a mimopracovních dní, ke kterému by se přihlíželo při generování docházky a při vytvoření absence.

4 Řešení zadaných úkolů

4.1 Webové stránky rukov.cz

4.1.1 Použité technologie

Jako první uvedu technologie, které jsem při řešení úkolu využil.

HTML

HTML je základním stavebním kamenem pro vývoj webových stránek. Jedná se o značkový jazyk, který se používá k vytváření obsahu a struktury webových stránek. Dříve jazyk HTML sloužil i k formátování vzhledu, ale v současnosti je dobrým zvykem k tomuto využívat kaskádových stylů, které vzhled od obsahu stránek oddělují a vytváří tak nezávislou vrstvu.

Název HTML je zkratkou z HyperText Markup Language, tedy hypertextový značkový jazyk. Díky hypertextovým odkazům můžeme vzájemně propojovat různé webové stránky. Pomocí některých speciálních značek, tzv. tagů, lze přiřazovat jednotlivým blokům textu význam. Můžeme tak část textu určit jako nadpis apod.

Jazyk HTML vychází z původního univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language). Vznikl v roce 1990 ve Švýcarsku a dále se vyvíjel v závislosti na nejpoužívanějších prohlížečích. Po verzi 4.01 došlo k zastavení vývoje a předpokládalo se, že HTML bude nahrazeno XHTML. K tomu však nakonec nedošlo a jeho vývoj byl obnoven. Aktuálně se pracuje na nové verzi 5. Ta přináší nové elementy pro sémantiku webu, ale také pro grafiku, multimedia a formuláře. Začíná také s podporou některých API, například pro geolokaci. [3][4]

CSS

Jedná se o jazyk, který určuje vizuální podobu dokumentů napsaných v jazycích HTML, XHTML či XML. Zkratka CSS pochází z anglického Cascading Style Sheets, tedy kaskádové styly. Postupný přesun definice vzhledu z HTML dokumentů do CSS souborů byl správný jak z důvodu logického oddělení designu od struktury a obsahu stránky, tak z důvodů přehlednosti a náročnosti vývoje. Stylování tak nemusí být upravování ve všech jednotlivých stránkách, ale postačí pouze propojit CSS soubory a dané stránky.

Nejnovější verzí je v současné době CSS3. Ta jde samozřejmě ruku v ruce s novým HTML5. Vybranými novinkami jsou animace, gradienty, nové selektory a pseudoselektory či tzv. media queries, díky kterým můžeme vzhled stránky přizpůsobovat dle zařízení uživatele. Například dle rozlišení displeje. [5][6]

JavaScript

JavaScript je užíván k programování chování webových stránek. Můžeme jej charakterizovat jako skriptovací, objektový jazyk založený na prototypch. Z toho vyplývají i některé jeho další rysy jako dynamický typový systém. Spadá mezi skupinu interpretovaných jazyků. Z jeho vlastností vyplývá, že je jednoduchým pro naučení a taktéž, že podporuje rychlý vývoj

kódu. Opravdu k mnoha změnám postačí pouze sepsání několika řádků, a proto se stal tak velmi populárním.

JavaScript může být zpracováván jak na straně klienta, tak přes vývojové prostředí Node.js také na straně serveru. Nicméně nejčastější využití je právě pro kód spouštěný u klienta, zatímco u serveru se uchytily jazyky jako PHP aj.

Jazyk vznikl v roce 1995 a stal se ECMA standardem o dva roky později. Jeho oficiální název zní ECMAScript. [7][8]

jQuery

jQuery je JavaScriptovou knihovnou, která se uchytla a je velmi využívána při vývoji webových stránek a aplikací, protože značně zjednodušuje a urychluje jejich implementaci. Ne nadarmo je sloganem knihovny "write less, do more". Nejčastěji se používá ve spojitosti s manipulací s CSS nebo reprezentací DOM. Umí také spolupracovat s AJAXem nebo HTML událostmi. Všechnu tuto práci ulehčuje a je tak možno psát ještě kratší kódy. I proto se jedná o nejpulárnější JavaScriptový framework, který je podporován ve všech hlavních prohlížečích.[9]

Bootstrap

Jedná se o velmi oblíbený front-end framework pro rychlý a snadný vývoj, který pracuje s technologiemi prezentační vrstvy, jakými jsou HTML, CSS a JS. Jeho velkou předností je, že díky němu můžeme velmi jednoduše vytvořit web s moderním vzhledem i chováním, který je navíc plně responzivní, čehož je dosaženo systémem mřížek.

Bootstrap poskytuje designové vzory pro typografii, formuláře, tabulky či tlačítka, jimiž je možno celou stránku sladit. Stejně tak lehko je možno využít pokročilejší prvky jako různé animované přechody mezi bannery nebo vyskakovací okna a nápovědy. Výsledný efekt na vzhledu tak může být opravdu značný. Momentálně Bootstrap vychází ve verzi 4, která je podporována všemi moderními prohlížeči. [10]

PHP

PHP je skriptovací, objektový programovací jazyk, se kterým se při tvorbě dynamických webů setkáme nejčastěji. Jeho podíl mezi jazyky určených pro webové aplikace, jejichž skripty jsou spouštěny na straně serveru, činí téměř 80%. Drží si tak velký náskok před dalšími technologiemi v pořadí, jíž je ASP.NET. Využívá se i pro velké projekty typu Facebook či WordPress.

PHP je rekurzivní zkratkou z PHP: Hypertext Preprocessor. První verze jazyka pochází z roku 1994. Pro svůj chod vyžaduje webový server. Nejčastěji se využívá tzv. balíček LAMP, který představuje zkratku s odkazem na 4 různé technologie používané pro provoz dynamických webových stránek. Těmito technologiemi jsou operační systém Linux, webový server Apache, systém řízení báze dat MySQL a právě jazyk PHP. Samotný jazyk PHP je však kompatibilní i s jinými HTTP servery jako například IIS. Také může běžet i na jiných platformách, například pod Windows. Takový balíček se pak označuje jako WAMP. A do třetice PHP podporuje také mnoho různých databází. Nespornou výhodou těchto balíčků je, že všechny jeho technologie jsou vydávány s open source licencí.

Mezi charakteristiky jazyka patří i dynamický typový systém a další rysy, díky kterým se lze jazyk velmi rychle naučit a dále v něm i rychle a jednoduše vyvíjet. [11][12][13]

CMS

CMS je zkratkou z Content Management System, což je obecné označení pro technologii určenou ke správě obsahu webových stránek zákazníkem. Jedná se o webovou aplikaci a také o pojmenování firemního frameworku, který je napsán v jazyce PHP. Díky tomuto konceptu je zákazník při potřebě úprav webových stránek schopen většinu změn provést sám, aniž by potřeboval znát jakýkoliv jazyk. Přes prostředí aplikace zvládne zamýšlené změny "naklikat" jako přes editor. Nejznámější CMS technologií je dnes WordPress.

Systém, se kterým jsem pracoval, je v základu rozdělen do sedmi částí, jimž odpovídají záložky, které klient spatří při vstupu do administrátorské sekce webových stránek. Jedná se o části nazvané Struktura, Knihovna, Moduly, Styl, Předlohy, Katalog textů a Uživatelé.

Sekce Struktura je důležitou částí, jež definuje stromovou hierarchii webových stránek, podle které se generuje na webových stránkách hlavní menu a další navigační prvky. Zde tedy vytváříme nové stránky či mažeme stávající. Také zde definujeme různé jejich parametry. Jedním z hlavních je typ stránky, kde vybíráme, zda se jedná o statickou nebo dynamicky vygenerovanou webovou stránku. Dalšími možnostmi jsou odkaz pro přesměrování na jinou adresu a externí zdroj.

Nesmím opomenout klíčový prvek, který je samotnou podstatou CMS systému, tedy obsah. Také ten je součástí záložky Struktura. Pro soulad s tím, co jsem již zmínil dříve, ani zde se klient neseťká s žádným jazykem, a tak pro úpravu obsahu stránky nemusí znát HTML, ale obsah vytváří, upravuje a formátuje přes editor. Co je obsahem je však ovlivněno typem stránky. Pokud se jedná o dynamickou stránku, je jím tzv. modul, který doplníme o data a tím vytvoříme seznam objektů (např. produktů). Samozřejmě i dynamicky vygenerovaný obsah lze doplnit o statické informace. Struktura CMS systému je záložka, která nám toho nabízí ještě daleko více, například zde můžeme řešit také SEO optimalizaci pro vyhledávače, pro zevrubnou představu si myslím, že je však její popis postačující.

Pod názvem Moduly se ukrývá stěžejní část pro tvorbu dynamického obsahu. Jednotlivé moduly můžeme připodobnit k třídám z objektově orientovaného programování či k entitám využívajících se v databázových systémech. Atributy entity zde odpovídají tzv. komponentám modulu. Ty mohou být různého typu, a tak není žádný problém vytvořit si například vlastní galerii či seznamy souborů ke stažení. Pokud navíc rozumíme alespoň základům HTML, jsme schopni měnit i design a způsob zobrazení modulů, a to pomocí kombinace specifických konstrukcí nahrazujících znalost programovacího jazyka.

V záložce Knihovna klient spravuje obrázky a ostatní dokumenty, které si přeje zobrazit nebo vystavit. Část Styl umožňuje dodatečnou změnu designu webu přes CSS stylování. Nicméně při vývoji je vzhled a layout webové stránky vyvíjen na míru zákazníkovi, proto není následně nucen vybírat z šablon a stylováním se nezaobírá. Předlohy jsou částí, kde si můžeme nadefinovat a uložit obsah, který lze přiřadit více stránkám, kde jej poté můžeme ještě doupřávit. V katalogu textů nalezneme textové řetězce, které sice nejsou součástí obsahu stránky, ale

nacházejí se v jejím layoutu, například v jejím zápatí. Zákazník je tak schopen si sám řídit obsah i ostatních komponent na webu. Poslední záložka s názvem Uživatelé je určena ke správě oprávnění v systému CMS.

Dle mého názoru se jedná o systém, se kterým se pracuje velmi jednoduše a pro správu obsahu je tak vyhovující. Obecné výhody CMS vychází z principu správy obsahu klientem. Ten si může web libovolně upravovat nezávisle na jiné organizaci vlastními kapacitami bez prodlev, bez dodatečných finančních nákladů a přesně podle svých potřeb. [14]

4.1.2 Řešení

Řešení zadaného projektu by se dalo rozdělit na dvě paralelní linie, kdy pro tvorbu obsahu či struktury webu bylo zapotřebí využít přímo administrátorského prostředí webové aplikace CMS. Naproti tomu layout stránky jsem musel definovat pomocí HTML kódu zapsaného v PHP v tzv. custom části frameworku. V této části jsem také vytvářel CSS a JavaScriptové kódy pro implementaci navrženého designu.

Jako první jsem vytvořil stromovou strukturu webových stránek s jejich vlastnostmi v administrační sekci Struktura systému CMS. Poté jsem se zaměřil na jejich design s tím, aby co nejvíce odpovídali předloze. Pro splnění podmínky responzivity jsem využil nástroje Bootstrap.

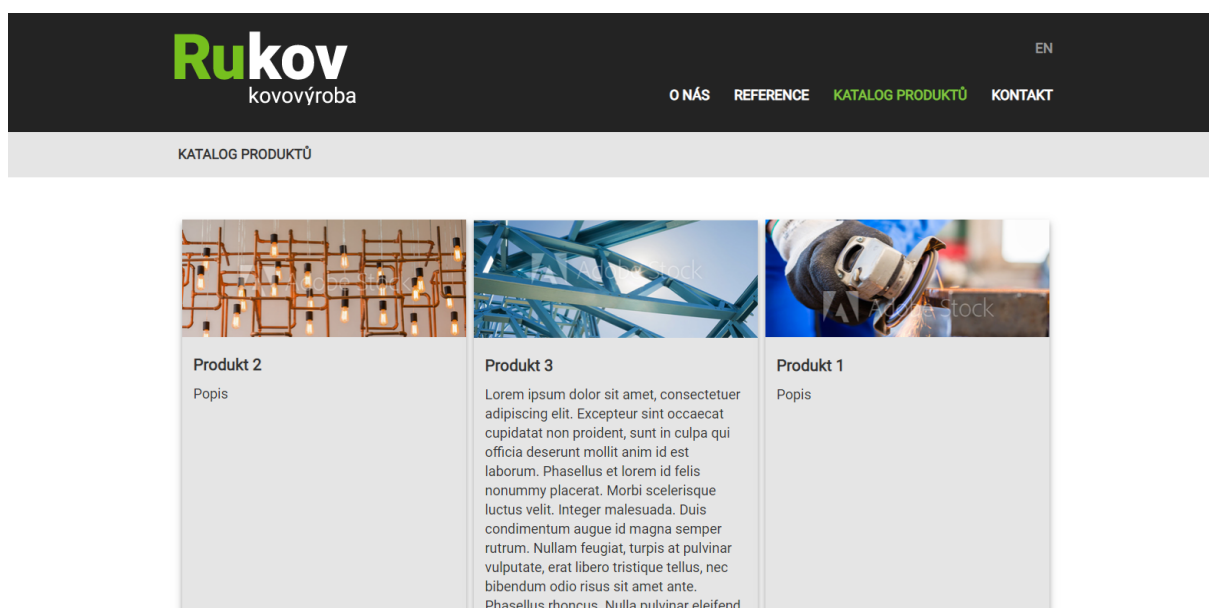
Designovou část jsem započal definicí obecného rozložení všech stránek v PHP. K tomu jsem využíval dvě funkce k tomuto sloužící. Jednu pro vygenerování hlavičky a druhou pro zápatí stránek. Právě mezi tyto funkce je vklíněn obsah doplněný zákazníkem. Ze zadání jsem si vypreparoval logo, to doplnil o přepínač do anglické verze stránky, menu a název podstránky a vše nastyloval dle požadavků. Menu, jehož položky byly na velkých obrazovkách zobrazeny vedle sebe, jsem pro lepší ovládání na malých displejích nechal pod sebou, k tomu jsem potřeboval využít tzv. media queries z nové verze CSS3. Poté bylo potřeba zaměřit se na banner hlavní stránky, který měl být ještě součástí hlavičky. Aby měl zákazník možnost si banner kdykoliv změnit, bylo zapotřebí využít záložky Knihovna, kde jsem jej nahrál a odkud jsem jej pak pod zvoleným názvem ve funkci pro layout nahrával.

V zápatí stránky se kromě informací o provozu webu měly nacházet také kontaktní údaje. Opět z důvodu, aby si klient mohl tyto údaje kdykoliv sám změnit, bylo zapotřebí tyto textové řetězce zakomponovat do administrace CMS. To je možno v rámci Katalogu textů. Textům jsem přiřadil název, přes který jsou pak v custom části frameworku nahrány do globální proměnné a dále využívány v layoutu.

Nakonec jsem se ještě speciálně zabýval dynamickými stránkami, pro které bylo zapotřebí v administraci již vytvořit moduly, jelikož vyžadovaly speciální design, který bylo zapotřebí naimplementovat. V obou případech se jednalo o vzhled karty s obrázkem, názvem a případně i popisem. Jejich počet v jednom řádku byl ovlivněn šířkou stránky, jelikož jejich šířka byla nastavena dle zadání na pevno. Pomocí stylování jsem také zajistil, aby výška všech karet v jedné řadě byla vždy stejná dle nejvyšší z nich. Aby karty lépe vynikly a nabyly moderního vzhledu, tak jsem pomocí stínování vytvořil dojem 3D prostoru, z něhož vystupují. Ačkoliv tato korekce ze zadání nevycházela, tak jsme se následně s kolegou shodli na ponechání této

úpravy.

V průběhu řešení jsem také narazil na problém, který se týkal umístění zápatí, a to konkrétně toho, aby se zápatí zobrazovalo i při velmi krátkém obsahu až na samém spodku okna prohlížeče, a nikoliv přímo za dodaným obsahem. Toto jsem řešil pomocí JavaScriptu a s využitím knihovny jQuery. Zjistil jsem si výšku zápatí a podle ní mu pak nastavil absolutní pozici od konce stránky. Aby nedošlo k překrytí části obsahu bylo zapotřebí ještě stejnou výšku nastavit jako spodní vnitřní okraj elementu body přes CSS vlastnost padding-bottom. Tento postup jsem spouštěl ve funkci jak při načtení stránky, tak při změně velikost okna, kdy by se mohla výška zápatí změnit.



Obrázek 2: Ukázka webu rukov.cz - řešení katalogu produktů

4.2 Firemní intranet

4.2.1 Použité technologie

Taktéž zde bych nejprve představil technologie, jichž jsem při řešení upotřebil, a které jsem doposud nezmínil.

JavaScript

Při řešení úkolu jsem pracoval s JavaScriptem, jenž jsem již popsal výše. Zde bych chtěl tento popis ještě doplnit o JavaScriptový framework jQuery UI.

jQuery UI

jQuery UI je framework zaměřený na uživatelské rozhraní. Nabízí množství komponent a efektů pro různé oblasti webové stránky, a tak umožňuje vytvořit přívětivější prostředí pro uživatele. [15]

MySQL

Technologie MySQL je systémem pro řízení báze dat. Lze ji charakterizovat jako multiplatformní, rychlý, spolehlivý a jednoduchý nástroj, který je navíc volně šiřitelný. Využívá standardu SQL a její vývoj dnes spadá pod společnost Oracle. Pro její administraci se často využívá přístupu skrz webové rozhraní nástrojem PhpMyAdmin. Já jsem během své odborné praxe pracoval s programem MySQL Workbench. [16][17]

AJAX

Tato JavaScriptová technika umožňuje programátorovi načítat data ze serveru nebo je tam zasílat i po načtení webové stránky a poté části této stránky dle získaných dat aktualizovat, aniž by muselo dojít k jejímu znovunačtení. Podstatou toho všeho je XMLHttpRequest objekt. Ten je zasílán na server, který tento požadavek zpracuje a zašle zpět odpověď. Vše probíhá asynchronně, to znamená, že program nečeká na odpověď volaného skriptu, ale pokračuje dalším kódem. Teprve až je přijata odpověď, spustí se tzv. callback funkce. AJAX je zkratkou pocházející z pojmenování Asynchronous JavaScript and XML. [18][19]

CVS

CVS je zkratkou z Concurrent Versions System a označuje program pro zálohování různých verzí zdrojových kódů. Díky tomu můžeme zpětně sledovat provedené změny a případně je i zrušit. Interně si program uchovává pouze jednu kopii souboru a záznamy všech úprav. Pro poskytnutí souboru v konkrétní verzi pak dochází k rekonstrukci dle těchto záznamů. Další výhodou systému je, že podporuje souběžnou práci více programátorů na stejném projektu. [20][21]

Firemní PHP framework

Tento framework napsaný v jazyce PHP je určen spíše pro interní webové aplikace a tímto se liší od systému CMS, který se využívá pro webovou prezentaci. Nabízí podporu pro řešení mnoha častých problémů. Těmi může být různá evidence se základními CRUD operacemi a například filtrováním. Oporu poskytuje ale také pro práva, logy, emailing, export dat atd.

Intranet

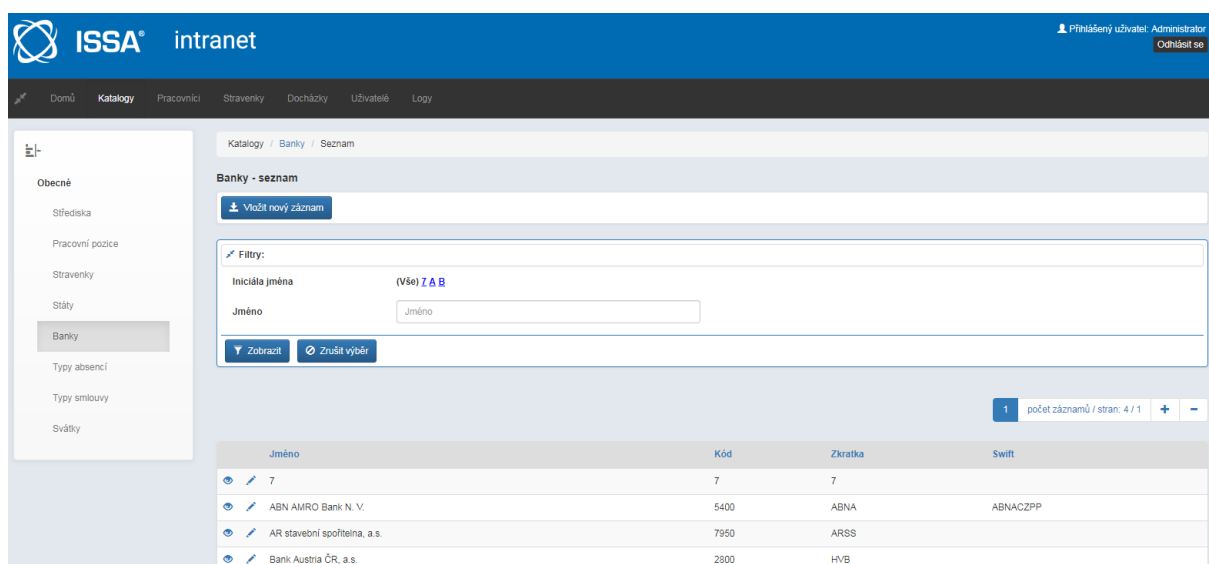
Pojmem intranet označujeme část počítačové sítě, která využívá stejné technologie jako Internet (např. rodina protokolů TCP/IP s protokoly HTTP, SMTP, atd.), ale narozdíl od něj je privátní. To se vyznačuje omezením přístupu, který je podmíněn přihlášením. V podstatě se jedná o interní webovou aplikaci, která slouží k zefektivnění různých vnitropodnikových procesů zvláště v oblastech administrace, komunikace či řízení podniku. Slouží také k usnadnění sdílení informací či jako rezervační systém. [22][23]

4.2.2 Řešení

Práci na projektu jsem započal tvorbou dvou číselníků v PHP frameworku. Prvního pro evidenci bank a druhého pro navazující seznam států. Tato tvorba spočívala ve vytvoření tzv.

view a dvou tříd pro každý z číselníků. Jedna z tříd reprezentuje předpis pro konkrétní záznamy bank či států. Tady dochází k popisu atributů objektů a také se zde nachází funkce pro generování popisů do logu historie a uložení a nahrání dat při práci s databází. Druhá z nich pak představuje celou stránku se správou těchto záznamů. V té jsem definoval například podobu seznamu, detailu, ale také validaci, požadované filtry nebo autocomplete komponentu pro přiřazení státu ze seznamu již existujících záznamů bance.

Zde jsem se zabýval také speciálním filtrem ze zadání (viz obrázek 3). Jako první jsem vytvořil funkci, ve které dojde k vygenerování podoby tohoto filtru. Ve funkci si nejdříve z databáze zjistím potřebná data a poté vygeneruji jeho HTML kód. Ten se skládá jednak z abecedně seřazeného výpisu iniciál názvů bank a pak také ze skrytého elementu vstupu formuláře, do kterého ukládám právě vybranou možnost. Po kliknutí na jednu z možností se spustí JavaScriptový kód, který tento skrytý vstup naplní dle výběru. Při odeslání formuláře je pak společně s ním zaslána také hodnota, dle které dochází k vyfiltrování příslušných záznamů.



Obrázek 3: Ukázka z intranetu - filtr iniciály jména

Dále jsem započal práci na evidenci stravenek. Oproti vytvořeným číselníkům tento modul vyžadoval komplexnější logiku s více funkcionalitami. Sumace na úrovni seznamu vyžadovala vytvoření a volání funkce s dodatečným dotazem do databáze, jelikož měla být provedena napříč stránkováním. Zároveň jsem zachoval obecnou logiku frameworku při generování dotazu do databáze, aby dotaz zůstal proveditelný za různých podmínek.

Největším problémem zadání tohoto modulu pro mě bylo hromadné vygenerování a vložení záznamů. Nejprve jsem na stránku se seznamem přidal vlastní formulář pro vyplnění roku s měsícem a tlačítko pro odeslání těchto dat. Dále jsem zajistil jeho spustitelnost na obou místech stránky, kde došlo k jeho vygenerování. Po stlačení tlačítka dojde k přesměrování na stránku s vygenerovaným seznamem záznamů pro vložení. Pro tuto funkcionalitu jsem ve frameworku vytvořil vlastní akci, která jakmile dojde k jejímu spuštění, načte z databáze všechny zaměstnance, kterým v zadaný měsíc a rok ještě nebyly vydány stravenky, a kteří jsou v zada-

ném období zaměstnanci firmy a pro ně výdajový záznam vygeneruje. Vygenerovaný seznam je pak předložen uživateli jako formulář, do kterého vyplní pro každého zaměstnance odpovídající počet vydaných stravenek. Také zatrhne záznamy, které chce opravdu uložit. Pro lepší ovládání jsem s pomocí JavaScriptu vytvořil zatržítka pro výběr všech záznamů, které jsem pro jeho obecnost využil i na několika dalších místech. Po stisknutí tlačítka pro uložení dojde k provedení další akce, ve které se uloží jak vygenerované záznamy, tak k nim odpovídající zápisy do logu historie.

Další netriviální záležitostí, kterou jsem řešil, bylo automatické nastavení typu stravenky po výběru zaměstnance ve formuláři pro vložení nového záznamu. Toho jsem docílil využitím techniky AJAX. Pro autocomplete komponentu frameworku jQuery UI jsem poupravil získávané hodnoty po výběru konkrétního zaměstnance o hodnoty Id a Value stravenky s ním spjaté. Pomocí těchto údajů ve formátu JSON jsem pak byl schopen nasimulovat automatický výběr odpovídajícího typu stravenky.

Pro potřeby emailingu a nastavení práv jsem využil zabudované funkce frameworku, díky kterým se jednalo o poměrně jednoduchou záležitost. V případě nastavení základních práv projekt využívá generátorů pro jejich vygenerování a přiřazení rolím. Pro uživatele s typem oprávnění zaměstnanec bylo zapotřebí, abych využil funkce pro omezení přístupu k pouze odpovídajícím záznamům. I pro toto jsem ve frameworku našel jistou oporu, která dle nastavení automaticky přidává do všech dotazů související podmínku.

Mým dalším úkolem byl docházkový systém. Požadavky na modul pro evidování absence byly obdobné jako ty dříve mnou již popsané. Například emailing nebo hromadné schválení absence pracovalo na stejných principech. Proto bych pozornost upřel rovnou na evidenci docházek.

Jistou zvláštností se ukázala stavba na různých úrovních systému. Ačkoliv na úrovni databáze jsem pracoval pouze s jednou tabulkou, v aplikaci se vyskytovaly již tři třídy z ní vycházející, které uživateli poskytly náhled na problematiku ve dvou agendách. Docházkách jednotlivců a pak také souhrnných docházkách všech zaměstnanců.

Tato specialita ale způsobila, že velkou část podpory ze strany frameworku nebylo možné využít, a tak jsem musel doimplementovat i například obyčejné CRUD operace. Při práci na detailu docházky jednotlivce pro daný rok a měsíc jsem se zabýval jak vygenerováním tabulky daného měsíce se zaznamenanou docházkou a absencí, tak také statistikou. V té jsou zaznamenány plánované a odpracované hodiny a absence v měsíci dle jednotlivých jejích typů. Zvlášť jsem se musel zabývat také upravováním docházky, jejím mazáním, a zvláště jejím vygenerováním společně se zaznamenáním do logu historie.

V obou zmíněných agendách se vyskytl požadavek na poskytnutí totožné funkcionality hromadného vygenerování a uložení docházky pro všechny aktivní zaměstnance. Jelikož jsem podobnou funkcionality vytvářel již pro modul stravenek, zapracoval jsem na jejím zobecnění. Takto jsem poměrně jednoduše vytvořil prostředí pro interakci s uživatelem v této záležitosti. Poté mi stačilo již jen naimplementovat logiku vygenerování docházky pro daného zaměstnance v daném roce a měsíci. Tu jsem se rozhodl vytvořit pomocí procedury hned při databázi a docílit tak určitého zrychlení.

Tuto proceduru (viz Výpis 1) bych nyní blíže popsal. Jejím vstupem jsou ID zaměstnance, rok a měsíc. Jako první si z modulu pracovníků zjistím pracovní dobu zaměstnance v jednotlivých dnech týdne. Protože se v MySQL nevyskytuje podpora pro pole proměnných, tak si získané hodnoty dočasně uchovávám v tabulce. Pokud je danému zaměstnanci opravdu nějaká pracovní doba přiřazena, pak procházím jednotlivé dny měsíce a testuji, zda se nejedná o den pracovního volna. Pokud tomu tak není a pro daný den má zaměstnanec nastavenou pracovní dobu, pak do databáze vkládám nový záznam o docházce v daný den. Nakonec zkontroluji, zda byl do tabulky vložen alespoň jeden záznam. Jinak do ní přidám prázdný záznam, a to z důvodu abych označil, že došlo k vygenerování docházky pro zaměstnance, pouze je zcela prázdná. Takovou docházku si pak lze v systému libovolně doplnit o vlastní hodnoty.

```

1 CREATE PROCEDURE `CreateAttendanceForEmployee` (IN p_id_empl_employee INTEGER,
2 IN p_year INTEGER, IN p_month INTEGER)
3 BEGIN
4 DECLARE v_weekday_shorcut CHAR(2);
5 DECLARE v_date DATE DEFAULT CONCAT(p_year, '-', p_month, '-1');
6 DECLARE v_hours DECIMAL(3,1);
7 DECLARE v_record_inserted BIT DEFAULT FALSE;
8 DECLARE v_jmeno VARCHAR(50);
9 DECLARE v_prijmeni VARCHAR(50);
10
11 DROP TABLE IF EXISTS pracdoba;
12 CREATE TEMPORARY TABLE IF NOT EXISTS pracdoba AS (
13 SELECT * FROM (
14 SELECT 0 as id, pracdoba_po as hodin FROM empl_employee
15 WHERE id_empl_employee = p_id_empl_employee
16 UNION SELECT 1 as id, pracdoba_ut as hodin FROM empl_employee
17 WHERE id_empl_employee = p_id_empl_employee
18 UNION SELECT 2 as id, pracdoba_st as hodin FROM empl_employee
19 WHERE id_empl_employee = p_id_empl_employee
20 UNION SELECT 3 as id, pracdoba_ct as hodin FROM empl_employee
21 WHERE id_empl_employee = p_id_empl_employee
22 UNION SELECT 4 as id, pracdoba_pa as hodin FROM empl_employee
23 WHERE id_empl_employee = p_id_empl_employee
24 UNION SELECT 5 as id, pracdoba_so as hodin FROM empl_employee
25 WHERE id_empl_employee = p_id_empl_employee
26 UNION SELECT 6 as id, pracdoba_ne as hodin FROM empl_employee
27 WHERE id_empl_employee = p_id_empl_employee
28 ) week
29 );
30 IF (SELECT COUNT(*) FROM pracdoba WHERE hodin > 0) > 0 THEN
31 WHILE MONTH(v_date) = p_month DO
32 SELECT hodin INTO v_hours FROM pracdoba WHERE id = (SELECT WEEKDAY(v_date));
33
34 IF v_hours > 0 && (SELECT COUNT(*) FROM cis_svatky WHERE datum = v_date) = 0
35 THEN
36 INSERT INTO doch_dochazka(datum, hodin, id_empl_employee)
37 VALUES (v_date, v_hours, p_id_empl_employee);
38 SET v_record_inserted = TRUE;
39 END IF;
40
41 SET v_date = DATE_ADD(v_date, INTERVAL 1 DAY);
42 END WHILE;
43 END IF;
44 IF !v_record_inserted THEN

```

```
45         INSERT INTO doch_dochazka(datum, hodin, id_empl_employee)
46         VALUES (CONCAT(p_year, '-', p_month, '-1'), 0, p_id_empl_employee);
47     END IF;
48 END
```

Výpis 1: Procedura pro vygenerování docházky zaměstnance v daném roce a měsíci

Souhrnné docházky byly do jisté míry obdobou těch určených pro jednotlivce jen bylo potřeba více dbát na kvalitu a přehlednost zobrazovaných informací, jelikož docházka konkrétního zaměstnance se nyní zobrazovala celá v jednom řádku i společně se statistikou. Po zaevidování docházky všech pracovníků se tak v tabulce nacházelo velké množství informací. Pro lepší přehlednost jsem musel lehce změnit stylování těchto dat.

Navíc stejnou tabulku mělo být možno vyexportovat do formátu PDF. K tomu jsem využil nástroje třetí strany nazvaného TCPDF. Po zprovoznění nástroje jsem mu předal tabulku zapsanou v HTML kódu a výstupem mi ihned byla totožná tabulka, ale ve formátu PDF. Tu jsem již pouze doplnil o správné rozměry a poté jsem provedl pár dalších změn pro vhodnost k tisku a tímto byl požadavek splněn.

Tím byla implementační část projektu dokončena a přešlo se k fázi testování. V té jsem musel reagovat jak na případné chyby, tak na další pozměňovací požadavky či požadavky na doplnění dalších funkcionalit. Zde se například ukázala potřeba existence katalogu se svátky a dny pracovního volna pro vhodnější generování docházek.

5 Znalosti či dovednosti získané v průběhu studia a uplatněné v průběhu odborné praxe

Je vhodné abych neopomenul zmínit znalosti a dovednosti, které jsem za své studium nabyl, a které mi byly během mé odborné praxe nápomocny. Ty bych chtěl přiřadit k jim odpovídajícím předmětům, se kterými jsem se setkal.

Důležitými pro mě byli znalosti, které jsem získal ohledně databází. V tom mi byly nápomocny zvláště předměty *Úvod do databázových systémů* a *Databázové a informační systémy*. Mnohokrát jsem při vývoji systému potřeboval získat netriviální data, které se z uložených dat musely nejdříve odvodit, nebo které se od nich odlišovaly v jejich formátu. Tyto znalosti mi pomohly problémy řešit pomocí složitějších SQL dotazů už na úrovni databáze. Tímto jsem mohl dosáhnout snížení množství přenášovaných dat a mírného zrychlení. Také mi poskytly potřebné znalosti ohledně rozšiřujících možností jazyka SQL, jakými jsou kupříkladu procedury.

Dalším databázovým předmětem, který mi podal potřebné informace, jež jsem během praxe využil, byl předmět *Databázové systémy*. A to informace jak ohledně fyzického návrhu databáze a dostupných strukturách pro optimalizaci systému, tak i ty, které se týkaly například hromadného generování a vkládání dat.

Asi nejvíce užitečných znalostí, které mi pomohly, jsem získal v předmětu *Vývoj internetových aplikací*. Ten probíhal paralelně s mojí odbornou praxí a byl zaměřen právě na stejná témata, jaká byla náplní mého prvního úkolu týkajícího se vývoje webových stránek. Potkal jsem se zde s mnoha technologiemi. Kromě HTML se jednalo například o CSS, JavaScript, AJAX či Bootstrap. Při studiu předmětu jsem se ale dozvěděl také o moderních trendech na poli designu webu.

Jistě mi přišel vhod i předmět *Vývoj informačních systémů* a jemu předcházející *Úvod do softwarového inženýrství*. Zde jsem se seznámil s teorií o vývoji informačního systému, ale taktéž se vzory, z nichž některé byly ve frameworku, který jsem používal, naimplementovány.

Předmět *Programovací jazyky a překladače* byl pro mne svým způsobem také důležitý. Ačkoliv jsem se v něm s programovacím jazykem PHP nesetkal, poskytl mi přehled o rysech různých jazyků. O jejich společných, a naopak specifických vlastnostech. Proto pak moje první setkání s jazykem PHP nebylo plné překvapení a mohl jsem se jej rychle doučit.

Závěrem bych chtěl uvést čtveřici předmětů, kterými jsou *Programování I*, *Programování II*, *Algoritmy I* a *Algoritmy II*. Ty tvořily základ pro samotnou dovednost programování. Především předmět *Programování II* ukázal nejen jak programovat, ale jak programovat správně.

Z dalších obecných dovedností, které jsem měl během studia možnost dále rozšiřovat bych uvedl například dovednost čtení v kódu. Ta pro mě byla stěžejní hlavně při seznamování s firemním frameworkem. Obecně jsem taky získal povědomí o systematickém vývoji software a jeho částí, jelikož vývoj nějakého systému byl častou náplní semestrálních projektů různých předmětů.

6 Znalosti či dovednosti scházející v průběhu odborné praxe

Zde bych uvedl znalosti a dovednosti, které mi během absolvování odborné praxe chyběly, a které jsem se tak musel například samostudiem doučit.

Ačkoliv si myslím že jsem v průběhu studia získal opravdu spoustu užitečných informací v oblasti databázových systémů, tak jsem se s jedinou zmínkou o MySQL setkal až na samém konci bakalářského studia. O to víc jsem byl překvapen, když jsem zjistil, že MySQL patří mezi tři nejpopulárnější systémy řízení báze dat společně se systémy Microsoft SQL Server a Oracle. [24] Naštěstí jsem jej však znal již z dřívější doby, kdy jsem s ním pracoval ve svém volném čase. Také mi pomohlo, že odlišnosti od jiných databázových systémů, se kterými jsem se setkal, nejsou samozřejmě až tak velké. Zbylé nedostatky jsem odstranil samostudiem.

Další znalosti, které mi chyběly se týkaly jazyku PHP, se kterým jsem se dříve nikdy neseťkal. Jistou oporou mi byla znalost některých vlastností z jiných programovacích jazyků. Příkladem budiž dynamický typový systém, který jsem znal z jazyku JavaScript. Považuji za škodu, že jsem se s PHP při studiu neseťkal, neboť jde evidentně o hojně využívaný jazyk pro tvorbu webových stránek.

Jako nedostatek pro mě byla také nedostatečná znalost webových technologií. Ač jsem absolvoval předmět *Vývoj internetových aplikací*, tak jsem nemohl do celé problematiky vstoupit hlouběji, protože předmět pojímal téma velmi zešíroka a na podrobnější seznámení nebyl prostor. Určitě bych ocenil předmět, který by poskytoval detailnější informace v této oblasti a jehož náplní by byl také jazyk PHP.

Jak jsem se zmínil již v úvodu, bylo jedním z důvodů pro vykonání mé bakalářské práce formou odborné praxe to, že jsem chtěl poznat, jak probíhá vývoj reálného informačního systému. Tyto zkušenosti mi vskutku chyběly a vlastně jsem je mohl získat jedine praxí. Brzy jsem poznal odlišnosti oproti semestrálním projektům z různých předmětů, kde bylo úkolem nějaký informační systém vytvořit a byly tak jakousi simulací reality. U komerčního projektu jsem se kupříkladu musel zabývat i těmi nejmenšími detaily, které bylo potřeba odladit pro bezproblémovou funkčnost systému. Můj další postřeh se týkal výběru tématu a tvorby i samotného zadání semestrálních projektů, ve kterém jsem měl vždy volnou ruku. Tímto se člověk vědomě i nevědomky mohl vyhnout problémů, na které by při implementaci zadání narazil. Pokud se vyskytl nějaký problém, mohlo být většinou jednodušší pozměnit zadání nebo detaily opomenout než se zabírat jejich řešením.

Nakonec bych uvedl ještě scházející znalosti o systémech používaných ke správě verzí. Ve firmě jsme k tomuto používali nástroj označovaný jako CVS. Nejedná se o příliš vyčerpávající téma, ale zmínku doplněnou o ukázkou práce i v jiných verzovacích systémech bych uvítal.

7 Závěr

7.1 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Celkově bych absolvovanou odbornou praxi zhodnotil velmi pozitivně, a to jak ve smyslu přístupu firmy, tak ve smyslu možnosti praxi takto vykonat v rámci bakalářské práce ze strany univerzity. Tento projekt odborných praxí je pro studenty jedinečnou možností, jak si vyzkoušet začlenění do pracovního procesu firmy nebo práci na reálných úkolech. Zvláště jsem tuto možnost ocenil jako student bez jakýchkoliv předešlých zkušeností se zaměstnáním v oblasti informačních technologií. Velice pozitivní je určitě také vliv na politiku zaměstnanosti budoucích absolventů fakulty. Jsem rád, že lidé ve firmě mi byli nápomocni při mém začlenění do společnosti.

Během praxe jsem nabyl mnoha nových zkušeností a znalostí. Setkal jsem se se spoustou technologií, které jsem dříve neznal a v těch ostatních jsem se zdokonalil. Zabýval jsem se technologiemi pro webovou prezentaci, ale i technologiemi určenými ke správě dat. Výstupem mé práce ve firmě jsou webové stránky pro zákazníka postavené na CMS systému a intranet ISSA založený na firemním PHP frameworku.

Literatura

- [1] *Webové stránky, eshopy, mobilní aplikace, SEO, PPC - ISSA CZECH* [online]. [cit. 2019-04-14]. Dostupné z: <https://www.issa.cz/>
- [2] *O nás - ISSA CZECH* [online]. [cit. 2019-04-14]. Dostupné z: <https://www.issa.cz/o-nas-620.html>
- [3] *Co je HTML | Adaptic* [online]. [cit. 2019-04-14]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/html/>
- [4] *HTML5 Introduction* [online]. [cit. 2019-04-14]. Dostupné z: https://www.w3schools.com/html/html5_intro.asp
- [5] *Co je CSS | Adaptic* [online]. [cit. 2019-04-14]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/css/>
- [6] *CSS Introduction* [online]. [cit. 2019-04-14]. Dostupné z: https://www.w3schools.com/css/css_intro.asp
- [7] *Co je JavaScript? - Learning the Web | MDN* [online]. [cit. 2019-04-14]. Dostupné z: https://developer.mozilla.org/cs/docs/Learn/JavaScript/First_steps/Co_je_JavaScript
- [8] *JavaScript Tutorial* [online]. [cit. 2019-04-14]. Dostupné z: <https://www.w3schools.com/js/default.asp>
- [9] *JQuery Introduction* [online]. [cit. 2019-04-14]. Dostupné z: https://www.w3schools.com/jquery/jquery_intro.asp
- [10] *Bootstrap 4 Get Started* [online]. [cit. 2019-04-14]. Dostupné z: https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp
- [11] *Historical trends in the usage of server-side programming languages for websites, April 2019* [online]. [cit. 2019-04-14]. Dostupné z: https://w3techs.com/technologies/history_overview/programming_language
- [12] *PHP 5 Introduction* [online]. [cit. 2019-04-14]. Dostupné z: https://www.w3schools.com/php/php_intro.asp
- [13] *PHP Introduction* [online]. [cit. 2019-04-14]. Dostupné z: https://www.tutorialspoint.com/php/php_introduction.htm
- [14] *Co je CMS | Adaptic* [online]. [cit. 2019-04-14]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/cms/>
- [15] *JQuery UI* [online]. [cit. 2019-04-14]. Dostupné z: <https://jqueryui.com/>
- [16] *PHP: MySQL Database* [online]. [cit. 2019-04-14]. Dostupné z: https://www.w3schools.com/php/php_mysql_intro.asp
- [17] *Co je to databáze MySQL? - BEST-HOSTING.cz* [online]. [cit. 2019-04-14]. Dostupné z: <https://best-hosting.cz/cs/napoveda/co-je-to-database-mysql>
- [18] *AJAX Introduction* [online]. [cit. 2019-04-14]. Dostupné z: https://www.w3schools.com/js/js_ajax_intro.asp
- [19] *Lekce 1 - Úvod do AJAXu* [online]. [cit. 2019-04-14]. Dostupné z: <https://www.itnetwork.cz/javascript/ajax/uvod-do-ajaxu>
- [20] *CVS pro každého (1) - základy - Root.cz* [online]. [cit. 2019-04-14]. Dostupné z:

<https://www.root.cz/clanky/cvs-pro-kazdeho-zaklady/>

[21] *Co je to CVS? - IT Slovník* [online]. [cit. 2019-04-14]. Dostupné z:

<https://it-slovník.cz/pojem/cvs>

[22] *Co je to Intranet? - Správa.sítě.eu* [online]. [cit. 2019-04-14]. Dostupné z:

<https://www.sprava-site.eu/intranet/>

[23] *Co je Intranet | Adaptic* [online]. [cit. 2019-04-14]. Dostupné z:

<http://www.adaptic.cz/znalosti/slovnicek/intranet/>

[24] *DB-Engines Ranking - popularity ranking of database management systems* [online]. [cit. 2019-04-14]. Dostupné z: <https://db-engines.com/en/ranking>